Title	詰め将棋問題の自動生成アルゴリズムに関する研究 [課題研究報告書]
Author(s)	石飛，太一
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11341
Rights	
Description	Supervisor:飯田 弘之，情報科学研究科，修士

課題研究報告書

詰め将棋問題の自動生成アルゴリズムに関する研究

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

石飛 太一

2013年3月

課題研究報告書

詰め将棋問題の自動生成アルゴリズムに関する研究

指導教官　飯田弘之 教授

審査委員主査　飯田弘之 教授
審査委員　池田心 准教授
審査委員　長谷川忍 准教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1110005 石飛 太一

提出年月: 2013 年 2 月

概 要

1950 年ごろに Shannon によって Minimax[15] 法が提案されてから 60 年の間にゲーム研究は様々な進歩を遂げた．Shannon がチェスをコンピュータにプレイさせせようとしたのと同じように，日本ではチェスよりも馴染みの深い将棋をコンピュータにプレイさせせようと様々な試行錯誤が繰り返されてきた．その結果，現在ではプロ棋士と渡り合えるような AI の開発に成功するなど研究は発展し続けている．AI が強くなる一方で，ゲーム研究の間では AI を強くする以外のことにも注目するようになってきた．それが面白さ，エンターテイメントに着目した研究である．人間の面白いとは何か，面白いと思わせるにはどうすれば良いかという疑問が生まれ，現在様々な実験や解析が行われている．

詰め将棋は，将棋の局面を模したパズルである．実践的で実際の詰め将棋にも表れるような局面を利用することもあれば，人工的に普通ではありえないような局面を作り，パズルとすることもある．精巧に作られた詰め将棋は，解く人間に難しく面白いという独特の感情を持たせる．パズルである詰め将棋は，将棋の中から面白い部分を抽出した存在なのではないだろうか．詰め将棋についての研究は詰め将棋を解くという研究と，作るという研究がある．詰め将棋を解くという研究はここ数年で飛躍的に進歩し，現在解けない詰将棋問題はほとんど存在しない．現在は，詰め将棋より難しい必至問題を解く研究に移行するなど，一つの終着点にたどり着いた状態となっている．一方で詰め将棋を作るという研究は，あまり盛んではない．現在存在するいくつかの詰め将棋生成手法では，確かに詰め将棋を作ることはできるが，人間が作ることができるような長手数問題や，面白さを追求した問題の作成などは難しい．詰め将棋に関する研究では，これらの課題に取り組む必要があると考え，本論文では特に面白さについて調査を行っている．

詰め将棋は，解くことで面白さを実感できるが，実感できる面白さには問題によって差がある．解けるが特に何の感情もわかず，一問題として終わってしまう詰め将棋がある一方で，名作とたたえられ何年も記録として残る詰め将棋もある．これらの差は一体どこから来るのであろうかという疑問と共に，この差について調べれば，人間に面白いと感じさせている正体が何か分かるのではないかと考えた．同じように考えたと思われる先行研究では，小山らによって，人間が詰め将棋を評価する際どのように見ているかについて研究がなされた [14]．この研究では評価の高い詰め将棋と一般問題群とについて，詰め将棋を構成する要素ごと傾向の違いを解析している．その結果，どのような詰め将棋ならば評価が高くなるのかという一定の答えを得た．しかし，この手法ではある詰将棋問題を見たとき，簡単にその問題の評価を下すことはできない．考慮すべき詰め将棋の要素の多さや，それらを数値化できないことによって点数をつけることが難しいことが理由である．

本研究では，この詰め将棋の面白さを単一要素で，なおかつ数値で表せないかについて，手法の提案と実験を行った．面白さを表すと考えられる数値には，証明数と反証数という要素を候補として挙げた．証明数・反証数は，Proof-Number Search[10] などで使用される探索を効率的に行うための数値である．これを詰め将棋問題において使用した際，

探索用ではなく別の見方ができないかと考えた．探索中の証明数・反証数を記録し，評価の高い詰め将棋と一般的な問題の詰め将棋についてデータにどのような違いがあるのかを解析した．その結果，証明数・反証数が評価の差によって違いが見られることが分かり，一定の関わりがあることを示すことができた．また，現在の詰め将棋創作手法で作成される問題の面白さについても調査を行い，その結果，野下 [12] による手法では大量に作品を作れば優秀な作品も作成できることや，広瀬ら [13] の手法では面白い問題を作成することは難しいことを示せた．今後，この関係をより詳細なものにできれば，新たな詰め将棋創作手法を提案できるのではないかと期待する．

目 次

第1章　はじめに

1.1　背景

詰め将棋は通常の将棋と異なり，与えられた局面から相手を詰ます手順を見つけるパズルとなっている．詳細なルールと解が一つに決まるという特性から，ゲーム研究の対象として研究されてきた．詰め将棋についての研究は大まかに分けて，詰め将棋を解く研究，そして作る研究の二つがある．詰め将棋を解くという研究はこの60年の間に，Proof-Number Search[10] や Depth-First Proof-Number Search[7] など探索手法の提案により飛躍的に進歩した．最近では，世界で最長とされる詰め手数とされる詰め将棋問題，Micro Cosmos[11] を解く事にも成功している．詰め将棋を解くという研究は一つの終着点に向かい，現在は必至問題といった詰め将棋を発展させたパズルを解く研究などすこしずつ対象が移っていっている状況にある．

それに対し詰め将棋を作るという研究はあまり進んでいないのが現状である．詰め将棋を作るという研究には，ランダムに駒を配置して詰め将棋を作る手法 [12] と，詰み局面から手を戻す逆算によって詰め将棋を作る [13] 手法がある．どちらも，一定の詰め手数を持った問題を作成できるが，人間が作成できる100手を超えるような詰め将棋の作成には至っていない．また，自動生成された問題は人が作ったものに比べ面白みが足りないという意見も存在する．このように，詰め将棋を作るという研究はあるものの，おもしろい詰め将棋の作成に焦点を当てたものや，長手数詰将棋問題の作成が可能なものは存在しない．

本研究ではこれらの問題に取り組むため，詰め将棋問題を解く際の証明数・反証数に注目し，これら値と詰め将棋問題の評価について調査を行った．証明数・反証数とは，詰め将棋問題を解く際に利用する Proof-Number Search で用いられる値である．本来ならこれらの値は，単なる探索用の値でしかないのだが，これを詰め将棋の評価と合わせてみたときどのような見方ができるのかについて実験・検討を行った．

1.2　論文構成

本論文の構成は以下の通りである．

第2章 詰め将棋概要

本研究で取り扱う詰め将棋のルールや特徴について解説する．

第 3 章 証明数探索
: 詰め将棋を解く際に利用する探索手法について解説する．また，探索手法で利用する要素についての説明も併せて行う．

第 4 章 詰め将棋創作アルゴリズム
: 現状詰将棋問題を創作するにあたり有力な二つの手法について解説する．また，それら手法の特徴と問題点についても説明する．

第 5 章 証明数を用いた詰将棋問題の特徴
: 詰将棋問題の評価を数値で行うため，証明数を用いて詰将棋問題の解析を行った実験について説明する．証明数を取得する手法や，証明数を評価の基準にする提案，実験結果の考察などを行った

第 6 章 おわりに全体のまとめと，今後引き続き必要な研究課題について示した

第2章　詰め将棋

2.1　詰め将棋とは

2.1.1　概要

詰め将棋とは将棋の終盤局面を模したパズルである．問題として与えられたある局面から相手を詰ませることを目的とする．このとき，将棋のルールに加え，詰め将棋専用のルールをいくつか用いるのが特徴である．有名なのは王手の連続というもので，開始から終了まで攻撃側は常に王手をかけなければならないというルールがあり，これは実際の将棋には存在しない．このような詰め将棋は，元来将棋の終盤力を磨くためのものであった．しかし，いつからかパズルとして完成度を上げるため，実際の将棋では見ることがないような状況を人工的に作り出すことで複雑かつ高度な問題作成を行うようになっていった．加えて，パズルとしての完成度を高めるためにいくつかの特別なルールが付加されると，実際の将棋から独立し，詰め将棋として新たな分野を確立した．

高度なパズルとしての詰め将棋問題は，人工知能分野の良き題材として研究されてきた．詰め将棋の解が厳密に１つに決まることや，問題によって難易度が大きく異なることは，探索手法を試すのには最適であった．現在では，詰め将棋の解を得るためにいくつかの探索手法か考案され，非常に効率的に問題を解くことが可能となっている．

2.1.2　例題

実際に詰め将棋の問題を図 2.1 に示す．

これは 3 手詰め問題の例題である．持駒の銀と盤上のと金を利用し，王手の連続で相手を詰ませることが可能である．この３手という手数は，詰むまでの相手と自分の手数の合計である．なお解答は▲を攻撃側の手，△を防御側とすると▲３二銀　△１二玉　▲２三と の 3 手である．

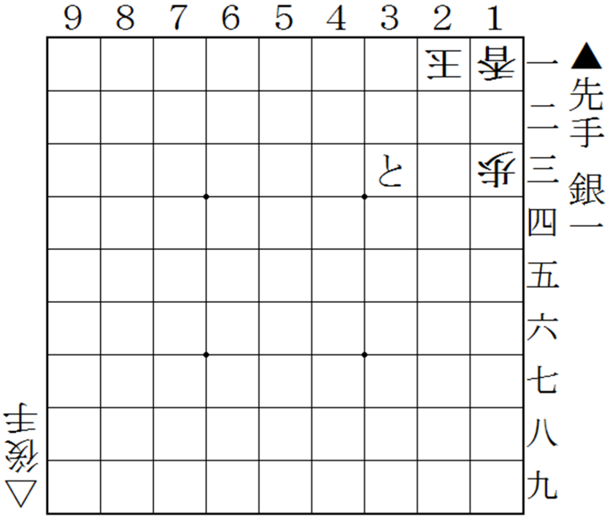

図 2.1: 3手詰め問題

2.2 詰め将棋のルール

ここでは詰め将棋のルールについて説明を行う．詰め将棋は基本的に将棋のルールを用いているがここでは割愛する．詰め将棋のルールは長い歴史の中で徐々に決まってきたものなので，明確な定義があるわけではない．従ってここでは野下 [3] にあるルールを参考にし，本研究において採用しているルールを示す．ルールには解く側のルールと問題作成側のルールの二つがあり，それぞれについて解説する．

問題を解く側のルール

以下に，詰め将棋を解く際に使用するルールについて示す．なお，ルールに登場する攻め方とは，玉を詰める側のことであり，玉方とは詰められる側のことを指す．

1. 攻め方が先手である
2. 攻め方は王手の連続で相手を攻めなければならない
3. 攻め方は予め与えられた持駒と，王手をしながら取った駒を使用してよい
4. 玉方は盤上と攻め方の持駒，玉将以外の駒を手駒として使用してよい
5. 攻め方は最短手順で玉方を詰めなければならない
6. 玉方は最長手順で詰みから逃げなければならない
7. 玉方は取られた後，詰め手順に変化を及ぼさないような合駒はできない

8. 玉方の逃げ手順で同手数の手順が複数ある場合は，攻め方に駒を与えない方を正解とする

基本的には王手の連続で攻め方が玉方を詰ますというパズルだが，これを多少複雑にしているのがルール 5,6,7 である．ルール 5 は，例えばある問題が 3 手で詰む場合と 11 手で詰む場合があるときは，3 手で詰む手順を採用するというものである．またルール 6 はルール 5 と同じく，ある手の後，1 手で詰む場合と 5 手で詰む場合があるときは，5 手で詰む手順を採用するという意味である．ルール 7 は少し複雑で，以下図 2.2 などにおいて適用されるルールである．

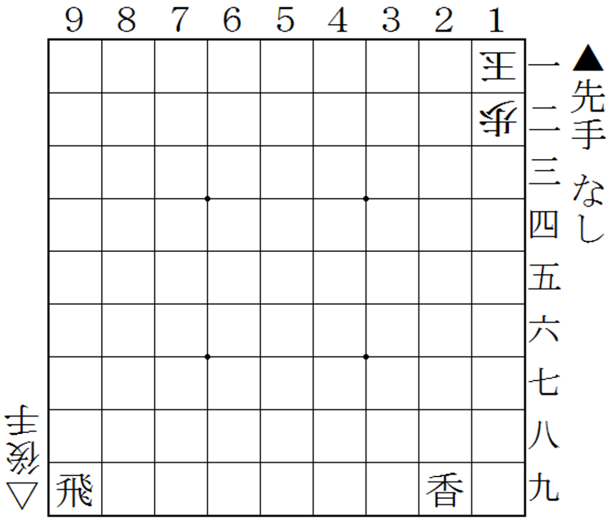

図 2.2: 無駄合いの発生する 1 手詰め問題

図 2.2 は，攻め方の▲ 9 一飛 の手で玉方が詰む 1 手詰め問題である．しかし，玉方が合駒を行うとこの問題は 15 手詰め問題となる．ルール 6 に従えば，この 15 手詰めが正しいように思える．しかし，この玉方の合駒は，最終的に飛車の攻撃を止められず，結局飛車によって玉は詰まされる．このように合駒の分だけのみ手順が伸びるような手を無駄合いと呼ぶ．ただしある作品において，合駒の 1 つが無駄合いかどうかという議論はたびたび行われており，一概に一つの合駒が無駄合いかどうかとは簡単には言えないのが現状である．

問題作成側のルール

問題を解く側に与えられるものとは別に，問題作成側に与えられるルールとしては以下のようなものがある．

- 最善（攻め方が最短手順，玉方が最長手順を選んだ）手順が一意に決まる

- 最善手順の最後で玉将が詰んだ際，攻め方の持駒がないようにする
- 盤上に飾り駒（その駒の有無が作品に関係しない駒）を配置しない

問題作成側が最も注意しなければならないのは，解答手順についてである．解答として用意した手順よりも短い手数，または同手数で別の解答手順が発見されると，作成された問題が余詰めありの不完全作として扱われてしまうのである．ただし，詰む直前の 1 手に限っては例外で，最後の 1 手が複数の方法で詰んだとしても，最終手余詰めと言われ不完全作とはならない．また，手順の中で飛び駒（香車，飛車，角など）を打つ際の位置が限定されない場合や，駒の成不成が自由である場合なども余詰めとはならない．ただし，この場合はキズと言われ，作品の完成度に悪い影響を与えるため注意が必要である．

2.3 詰め将棋と人工知能

詰め将棋と人工知能分野の関係は深く，詰め将棋が将棋と異なり解が一意に決まる，ゲーム木の最大規模が決まっている，局面の評価は詰む・詰まない・分からないかの三通りのみといった各種特性により長い間研究されてきた．詰め将棋の研究には二種類のものがあり，1 つは詰め将棋の解を得るという研究であり，もう 1 つは詰め将棋を作るという研究である．それぞれの研究の歴史と現状について簡単に説明する．

2.3.1 詰め将棋を解く事について

詰め将棋を解くという研究は当初 Shannon によるミニマックス法 [15] によって行っていた．例えば 5 手詰めの詰め将棋ならば，5 手までの手筋を全て探索すれば解を得られるのだが，これをミニマックス法に反復深化法を用いることで解を得ていたのが当初の研究であった．その内，Knuth の考案したαβ探索 [6] や評価関数の値を元に最良優先探索する手法 [1] などが考案され，徐々に解ける問題が増えていったが短手数の問題を解くのが限界であった．

1988 年に McAllester による共謀数探索 [8] が考案されると，この探索手法を元に L.V.Allis によって Proof-Number Search[10] が発表された．この手法は非常に画期的かつ効率的で，これまで解く事ができなかった詰め将棋が次々に解かれていった．発表から 2 年後にはこの探索方法を元にした瀬尾詰め [9] により世界最長の 1525 手詰め問題 Micro Cosmos[11] が解かれ，詰め将棋で解けない問題はほとんど無くなっていった．近年では長井による Depth-First Proof-Number Search[7] という Proof-Number Search よりも効率的な手法も考案されるなど，詰め将棋を解くという分野は 1 つの終着点に向かっているように思える．

2.3.2 詰め将棋創作について

詰め将棋を解くという研究が進む一方で，詰め将棋を作るという研究はあまりなされてこなかった．現在，大きく分けて二つの手法が存在しており，野下によるランダム配置を使用する手法 [12] と広瀬らによる逆算法を用いた手法 [13] が存在する．ランダム配置を使用する手法はランダムに配置した局面から駒を消すなどの操作をし詰め将棋を作成する．この手法では 25 手といった中編程度の問題作成ができるが，問題の完成度は初期配置のランダム性に依存していることや，局面を簡素化するため問題の難易度低下といった問題がある．逆算法を用いる手法では終局面，つまり詰んでいる局面から 1 手づつ戻すことで詰め将棋を得る．この手法では，作成できる詰め将棋の自由度が高いが探索範囲が広すぎるため，9 手詰め程度の短編問題作成が限界となる．

以上のように，現存する詰め将棋創作アルゴリズムには長手数問題の作成や，自由度の高い問題生成など課題は多く，まだまだ研究する余地のある分野だと私は考える．

第3章　証明数探索

3.1　証明数探索とは

証明数探索とは L.V.Allis によって 1994 年に考案された Proof-Number Search（以下 PNS）という探索手法である [10]．AND/OR 木と呼ばれるゲーム木に証明数・反証数という二つの概念を取り入れることで，効率的な探索が可能となる．その後，この PNS に深さ優先探索の概念を取り入れ，より効率的な探索となった Depth-First Proof-Number Search(以下 DFPN）という探索手法が長井 [7] によって 2002 年に考案された．ここでは本研究で使用したこの二つの探索手法について詳しく説明する．

3.2　AND/OR 木探索

始めに，証明数探索で用いられる AND/OR 木について説明する．図 3.1 に AND/OR 木の例を示す．

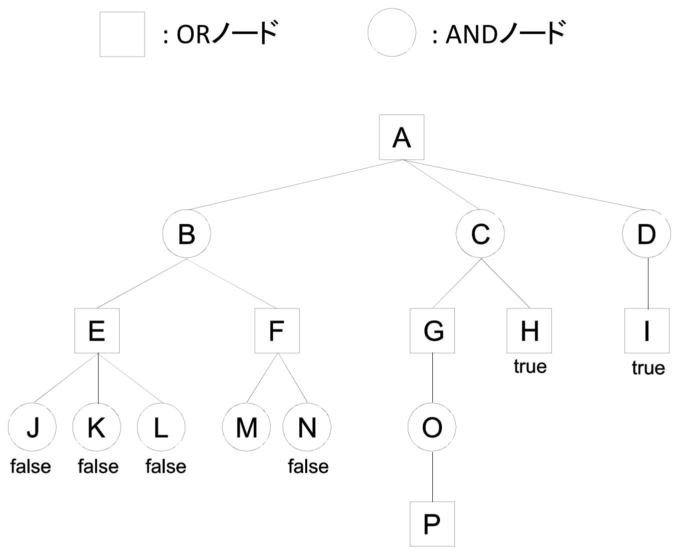

図 3.1: AND/OR 木の一例

AND/OR 木はゲーム木の一種で，図 3.1 のように AND ノード，OR ノードと呼ばれる二つのノードが交互に接続されたものを指す．それぞれのノードは True, False という二

つの値を持っており，値が確定しない部分は不明として扱われる．各ノードの値はそれぞれの子ノードによって決定され，例えば，ノード E の値は False, ノード C の値は不明，ノード D の値は True となっている．この値は，ブール代数の論理演算子である AND や OR と同じく，各ノードの子ノードに対しての AND 演算または OR 演算によって決定している．例えば，ノード F は子ノード M，N の値の AND 演算によって決定され，ノード M の値が不明であろうとも子ノード N が False の時点でノード F の値は False に決定される．同じく，ノード A の値はノード B，C の値にかかわらず，ノード D がノード I によって True と決まった時点で同じくノード A も True となるのである．このように，AND/OR 木では目的のノードの値を得るに当たって全てのノードの展開を行う必要はない．例えばノード A の値を知りたいだけならば，ノード D,I の展開のみで値を決定できる．

この AND/OR 木を詰め将棋に用いると，末端ノードにおける True とは，攻め方が玉方を詰めた局面であり，False とは玉を詰められなかった局面と置くことができる．例えば図 3.1 が詰め将棋の AND/OR 木だったと考えると，OR ノードは攻め方の手番であり，AND ノードは受け方の手番となる．すると，ノード E は攻め方の手番となりノード I,J,K は攻め方が選べる次の局面と表される．しかし，子ノード J,K,L は全て False であり，玉を詰ますことができない局面であるため，ノード E の値も False となる．ここで，例えば子ノード J の値が True，すなわち玉を詰ますことができる局面だったとすると，ノード E の値は OR 演算により True となる．このように，OR 演算の計算方法は，攻め方が次の局面を選べることを表しており，反対に AND 演算は攻め方が次の手を玉方に委ねなければならないことを表す．以上のようにして各ノードの値を決定していき，ルートノードの値が True になった場合は詰みへの手順が存在することを表し，False になった場合は不詰めであることを表す．

AND/OR 木は詰め将棋において，ノードの値決定方法とルートノードの値を確定させることで詰め将棋を解く事ができるというゴールを用意する．これに対し PNS および DFPN は，AND/OR 木のルートノード値を True と決定するのに，最も効率的なノードの展開および探索手法を提案する．

3.3 Proof-Number Search

Proof-Number Search は L.V.Allis[10] によって考案された探索手法である．AND/OR 木に対し証明数と反証数という二つの概念を取り入れ，最良優先探索によってルートノードの値を True に確定するための効率的な探索を提案する．本節では証明数および反証数の説明と，オリジナル PNS のアルゴリズムについて説明を行う．

3.3.1 証明数と反証数

証明数および，反証数とは以下の通りである．

証明数 あるノードの値を True に確定させるために，値が True になることを証明しなければならないノードの数

反証数 あるノードの値を False に確定させるために，値が False になることを証明しなければならないノードの数

図 3.2 に証

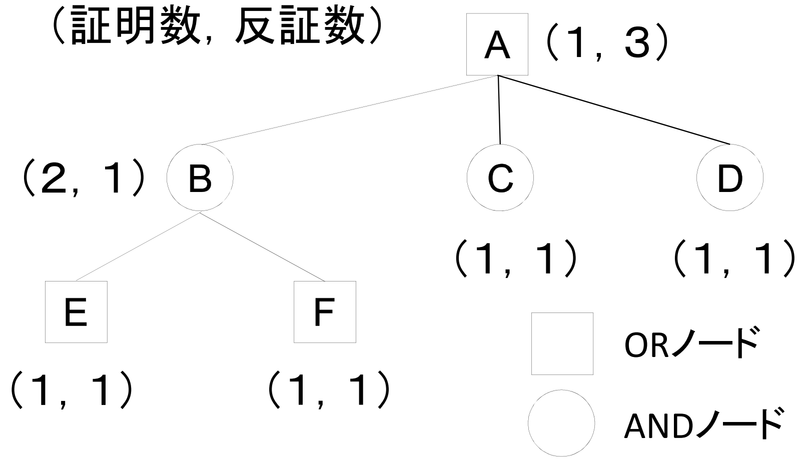

図 3.2: 証明数・反証数の例

ノード E などの末端ノードの証明数，反証数は 1 である．これは，ノード E の値を True または False に確定するのに調査しなければならないノードが，現在は自分のみだからである．ノード B の証明数は 2 となっているが，これはノード B が玉方の手番であるため，子ノード E,F の両方の値が True にならない限り，ノード B も True とはならないためである．反対に，ノード E,F のどちらかが False になればノード B の値も False になるため，反証数は 1 となる．ノード A の証明数は 1 であり，反証数は 3 だが，これはノード A は攻め方の手番であるため，子ノード B,C,D のうちどれか一つでも True になればよく，反対にノード A の値を False にするには子ノード全てが False になる必要があるためである．

証明数，反証数の値はゲーム木が展開されるたびに更新されていく．例えば，図 3.2 が完全な二分木であり，ノード C,D も同様に子ノードを 2 つ持っていた場合，ノード A の証明数は 2 に更新される．これはノード A の値が True になるにはノード B,C,D の子ノード全てが True であることを証明しなければならないためである．

証明数，反証数を数式で定義すると，あるノード N の値が True，False，または不明である場合，証明数（pn)，反証数（dn）はそれぞれ次のように表せる．

$$(N.pn, N.dn) = \begin{cases} (0, \infty) & \text{if N is True} \\ (\infty, 0) & \text{if N is False} \\ (1, 1) & \text{otherwise(N is unknown)} \end{cases} \tag{3.1}$$

証明数（反証数）が0とは，ノードの値がTrue（False）となったため，証明すべきノードがなくなったことを表す．反対に，証明数（反証数）が∞とは，そのノードをTrue（False）にするには∞のノードを証明する必要があることを表しており，言い換えれば値が確定したため，いくら証明しようとも値を変更することができないことを表している．

またノードNの値は，ノードNの子ノード群をCとすると次のように定義される．

$$(N.pn, N.dn) = \begin{cases} (\min_{c \in C} c.pn, \sum_{c \in C} c.dn) & \text{if N is OR node} \\ (\sum_{c \in C} c.pn, \min_{c \in C} c.dn) & \text{if N is AND node} \end{cases} \tag{3.2}$$

AND/OR木の証明数，反証数は末端ノードが展開され子ノード作られた際に計算され，展開したノードから順に親ノードまで値が更新される．このように決定された証明数，反証数を用いることにより効率的な探索を実現する．

3.3.2 アルゴリズム

PNSのアルゴリズムは次の通りである．

1. ルートノードの証明数，または反証数が0なら終了
2. N = ルートノード
3. Nが末端ノードでないなら次へ，末端ノードなら5へ
4. N = most_proving_node(N) とし3へ
5. Nを展開
6. 証明数，反証数を更新し1へ

most_proving_nodeはPNSにおける最良優先探索の対象となるノードを返す関数である．ノードNとその子ノードCに対するmost_proving_nodeは次のように定義される

$$\text{most\_proving\_node}(N) = \begin{cases} \arg\min_{c \in C} c.pn & \text{if N is OR node} \\ \arg\min_{c \in C} c.dn & \text{if N is AND node} \end{cases} \tag{3.3}$$

つまり，ORノードならば最も証明数の小さな子ノード，ANDノードなら最も反証数の小さな子ノードを指す．例えば，図3.2においてノードAのmost_proving_nodeはノードCまたはDである．PNSではmost_proving_nodeが複数ある場合は左側にあるものを優先するため，この場合はノードCとなる．この図ではノードCは末端ノードであるため，展開を行った後，ルートノードまで証明数・反証数の値を更新する．仮にノードCが

子ノードを持っている場合には，ノードAで行ったのと同様にmost_proving_nodeを調査し対象ノードへ遷移する．

この最良優先探索で探索しているのは，最も値を決定しやすいノードである．AND/OR木において説明したように，ANDノードおよびORノードの値は論理演算のように決定されるため，1つのノードの値がTrueまたはFalseに確定することにより，連鎖的に他のノードの値も確定する．この特性を生かし，一度に多くのノードの値を確定させようと探索するのがPNSである．例えば，証明数が1のORノードNについて，most_proving_nodeを探索し，末端ノードを展開した際に値がTrueと確定すれば，同時にノードNの値もTrueと確定する．最終的にルートノードの証明数または反証数が0（または∞）になれば探索終了である．証明数が0となり探索が終わった場合は，ルートノードの値はTrueであるため，詰みとなる手順が存在すると証明され，反対に反証数が0となった場合は，ルートノードの局面から相手を詰ますことはできないと証明される．

3.4 Depth-First Proof-Number Search

Depth-First Proof-Number Searchは長井[7]によって考案された探索手法である．PNSを改良した手法であり，AND/OR木に証明数と反証数を導入する部分は同じである．PNSとの最大の違いは深さ優先探索であること，証明数および反証数の閾値を導入していること，Transposition Tableを利用していることがある．Transposition Tableとは一度探索を行ったノードの情報を保存しておき，再度同じノードに探索を行った際に，証明数等の各種値を参照するためのテーブルである．ノード情報は局面のハッシュ値などを利用して登録されているため，初めて探索するノードでも，既に探索した局面ならノードを展開することなく値を決定できる．従って，使用するメモリの量や探索量を削減できるため，オリジナルのDFPNで使用されているが，本研究では採用していない．採用しない理由については5.2章を参照して頂き，ここでは本研究で採用しているアルゴリズムについて解説する．

3.4.1 アルゴリズム

DFPNではPNSと同じく証明数と反証数を利用するが，再帰的にプログラムを組むためノード毎に表すものが変わるϕ, δという変数を導入する．ノードNのϕ, δは以下のように表される．

$$N.\phi = \begin{cases} N.pn & \text{if N is OR node} \\ N.dn & \text{if N is AND node} \end{cases} \tag{3.4}$$

$$N.\delta = \begin{cases} N.dn & \text{if N is OR node} \\ N.pn & \text{if N is AND node} \end{cases} \tag{3.5}$$

これに伴い，ノードの値は次のように表される．

(i) ノード N が OR（AND）ノードで値が True（False）の場合

$$
\begin{aligned}
N.\phi &= 0 \\
N.\delta &= \infty
\end{aligned}
$$

(ii) ノード N が AND（OR）ノードで値が True（False）の場合

$$
\begin{aligned}
N.\phi &= \infty \\
N.\delta &= 0
\end{aligned}
$$

(iii) ノードの値が不明の場合

$$
\begin{aligned}
N.\phi &= 1 \\
N.\delta &= 1
\end{aligned}
$$

同じように，ノードの値を計算する式もノード N の子ノードを C とすると以下のように変更される．

$$
N.\phi = \min_{c \in C} c.\delta \tag{3.6}
$$

$$
N.\delta = \sum_{c \in C} c.\phi \tag{3.7}
$$

DFPN では各ノードに証明数と反証数の閾値という二つの値を新たに持たせている．ノード N の証明数の閾値を th_{pn}, 反証数の閾値を th_{dn} とすると以下のように表せる．

$$
N.th_{\phi} = \begin{cases} N.th_{pn} & \text{if n is OR node} \\ N.th_{dn} & \text{if n is AND node} \end{cases} \tag{3.8}
$$

$$
N.th_{\delta} = \begin{cases} N.th_{dn} & \text{if n is OR node} \\ N.th_{pn} & \text{if n is AND node} \end{cases} \tag{3.9}
$$

この閾値を用いたアルゴリズムを以下に示す．なお，ルートノードを r とする．

1. ルートノードの閾値を次のように設定する $r.th_{\phi} = \infty, r.th_{\delta} = \infty$
2. N = r とする

3. ノード N が末端ノードなら展開する．証明数，反証数を更新し次へ．

4. $n.\phi \geq n.th_{\phi} orn.\delta \geq n.th_{\delta}$ ならば 7 へ．そうでないなら次へ．

5. ノード N の子ノードの内最も小さい δ を持つ子ノードを N_c，二番目に δ が小さい子ノードを N_2 とする（最も小さい δ を持つ子ノードが他にもあるばあい，それを N_2 とする）．見つけた N_c について，閾値を以下のように設定する．

$$N_c.th_{\phi} = N.th_{\delta} + N_c.\phi - \sum N_{child}.\phi \tag{3.10}$$

$$N_c.th_{\delta} = \min(N.th_{\phi}, N_2.\delta + 1) \tag{3.11}$$

6. $N = N_c$ として 3 へ

7. $N = N$ の親ノード として 3 へ．N がルートノードの場合は終了

図 3.3 に DFPN の例を示す.

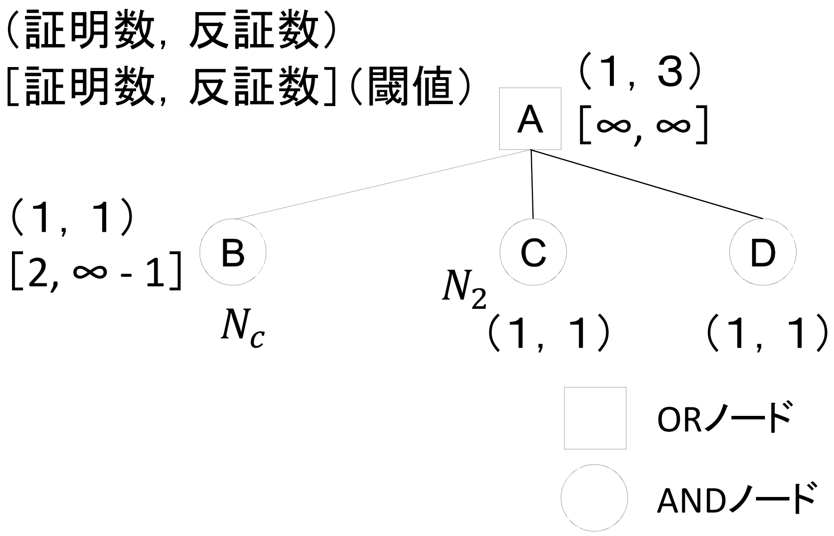

図 3.3: DFPN の一例

ϕ, δ という表現は動作を説明する上では分かりづらいため，ここでは証明数，反証数を用いる．まず，ルートノードの閾値は $r.th_{pn} = r.th_{dn} = \infty$ となる．ノード A の子ノードの内最も小さい δ，つまり AND ノードでは証明数を探す．最も小さい証明数を持つ子ノード N_c はノード B（全て同じだが，左側にあるノードを優先する），二番目に小さい証明数を持つ N_2 はノード C である．従って，次に探索するノードは AND ノードの B となり，閾値が設定される．証明数の閾値はノード N の証明数と N_2 の証明数 $+1$ の小さい方を設定するため，ここでは ∞ と 2 を比べることとなるため証明数の閾値は 2 となる．反証数の閾値はノード N の反証数に N_c の反証数を加え，さらに子ノードの反証数の合計を減算する．従って，計算は $\infty + 1 - (1 + 1 + 1) = \infty - 2$ となる．

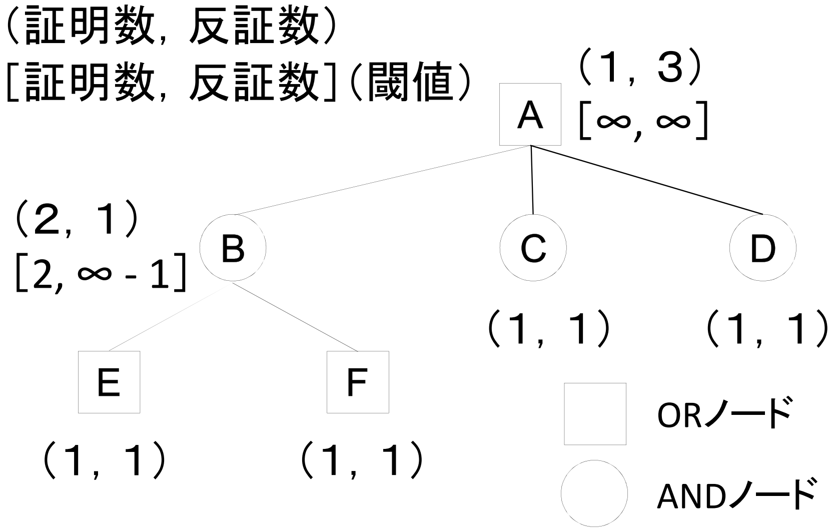

図 3.4: 図 3.3 においてノード B を展開した状態

次に，N_c を展開した図を 3.4 に示す.

ノード B を展開した結果，二つの子ノードを得た．これにより，ノード B の証明数は 2，反証数は 1 へと更新される．ここで閾値と比較を行うと証明数が閾値以上の値となっていることが分かる．従って，ノード B の探索をここで打ち切り，親ノードであるノード A に遷移する．この行程を繰り返すことで，いずれノード A の証明数または反証数が ∞ となり探索が終了する．反証数が ∞ となれば，詰みまでの手順が存在し，証明数が ∞ となれば不詰めの局面であることが分かる.

3.4.2 Proof-Number Search との違い

PNS と DFPN はいくつかの以下の部分で大きく異なる.

(a) 探索の方法

(b) 証明数，反証数の更新方法

(c) Transposition Table の使用による大幅なメモリ節約と高速化

まず (a) だが，これは PNS が最良優先探索，DFPN が深さ優先探索であることによる．厳密に言えば DFPN は深さ優先探索に反復深化を取り入れたような独特の動き方をし，ルートノードまで戻らず，あるノードの証明数または反証数が一定になるまで子ノードについて探索を繰り返す．この一定の値とは，同じ親を持つ子ノード群について，ある探索している子ノードが他の子ノードより証明しやすそうな間を表すものである．このようにPNS が末端ノードまでの探索を繰り返すのに対し，DFPN は一定の基準に従って中間ノードに留まりながら探索を繰り返す.

(b) については (a) と関連があり，PNS が末端ノードを展開した後ルートノードまで値を更新するのに対し，DFPN では現在見ているノードの値のみを更新する．従って，ルートノードの値は探索が戻ってくるまで更新されないため，PNS がルートノードの証明数または反証数を見れば現在の探索状況を把握できるのに対し，DFPN では難しいこととなる．

(c) については DFPN が深さ優先探索を行うためである．深さ優先探索の場合，最良優先探索とは異なり，ある局面を詳しく探索した結果が早い段階で分かる．これにより，Transposition Table に登録される情報量と質が向上し，別のノードを探索する際にも大きく役立つようになるのである．また，局面の優越関係を用いることにより，初めて探索する局面についても効率的に見ることができる．

以上のように PNS と DFPN にはいくつか違いがあり，基本的に DFPN の方が効率的かつ高速である．ただし，PNS と DFPN は数学的に等価であり，探索結果は PNS と DFPN で等しくなることが証明されている．

第4章　詰め将棋創作アルゴリズム

4.1　詰め将棋の創作について

詰め将棋を解くという研究についてはこの60年ほどで飛躍的に進歩し，ほとんどの詰め将棋を解く事に成功している．その一方で詰め将棋を作るという研究も行われてきており，代表的なものには野下[12]による手法と広瀬ら[13]の手法がある．どちらも一定の成果を得られてはいるが，好きな詰め将棋を自由自在に作るとはまだまだいかず，できあがる詰め将棋には限界があるのが現状である．ここでは，これら代表的な二つの手法について説明を行う．

4.2　野下によるランダム配置を用いた手法

4.2.1　概要

野下による手法ではランダムに配置した初期局面を利用し，駒の削除や変更と言った操作を用いることで詰め将棋を創作する．相手玉を含む駒を盤上および持駒としてランダムに配置を行い，詰み手順が存在するかどうかを検査する．詰み手順があった場合は，駒の操作と余詰めなどの検査を繰り返しながら詰め将棋としてのルールを満たすように局面を変更していく．

この手法では，主に13手から19手程度の問題を多く作成でき，20手を超える問題の作成にも成功している．数ヶ月で一定難易度の問題を1000問程度作成できるとされている．

4.2.2　アルゴリズム

野下のアルゴリズムを以下に示す．

1. 盤面にランダムに駒を配置する
2. 局面に詰め手順が存在するか検査する．存在しない場合は配置をやり直す
3. 局面の駒に対して操作（除去，減少，変換）を行う
4. 詰め手順があるか確認し，詰手数が操作前以上になっていれば局面を保存する

5. 3，4 のステップを可能な限り繰り返す

ステップ 3 において，駒の変更を行い，変更を行ったそれぞれの局面について詰み手順があるか，詰み手数が伸びたかを確認している．アルゴリズムを見れば分かるが，駒の変更には駒の追加や移動は含まれていない．駒の除去とは盤面から駒を取り除く操作，減少とは持駒を減らす操作，変換とは互換関係にある駒（香車を歩へ変換するなど）への変更を行う．この操作を論文では簡約関係にある局面への遷移として表され，元の局面のゲーム木から一部の枝を切り取った元局面の部分集合となる局面へ変換するという操作になる．駒の追加や移動は，元となるゲーム木を複雑に変更してしまうため互換性が失われる．互換性を保つことにより，最低でも最も始めに発見した詰み手順のある詰め将棋を作成するのである．

4.3 広瀬らによる逆算を用いた手法

4.3.1 概要

広瀬らの手法では予め用意した詰み局面を利用し詰め将棋を作成する．詰め将棋は正しい詰め手順を行えば最終的に玉方を詰ますことができる．この詰んだ状態から詰めて順を逆順に戻していけば元の詰め将棋を得ることができるはずである．これを利用し，広瀬らは詰み局面から一手戻した局面について検査を行い，詰め将棋のルールを満たしているものを採用していくことにより，詰め将棋問題を創作しようとした．

この手法では 3 手から 9 手程度の問題の創作に成功している．

4.3.2 アルゴリズム

以下に広瀬らの手法のアルゴリズムを示す．

1. 詰め上がりの局面を用意する
2. 逆算により，一手前の局面を生成する
3. 生成局面に余詰め，不詰めがないか確認する
4. 2，3 を可能な限り繰り返し，詰み手順の存在する局面を生成する
5. 生成した局面から余計な駒などを削除し，詰将棋問題を生制する

ステップ 2 で行っている一手前の局面の生成が逆算である．一手前の局面とは，例えば局面に歩だけが置かれているとした場合，歩が 1 つ後ろに下がった局面のことを指す．すなわち一手前の局面とは，現在の局面に合法的に遷移できる局面のことである．この手法で一手前の局面を全て作成し，詰め将棋のルールを引き続き満たしているものについて採用していくことによって詰め将棋を作成するのである．

4.4　既存手法の問題点

4.4.1　手数について

野下による手法では，20 手を超える詰め将棋を作成できる一方で，作成できる詰め手数の最長に限界があるという問題がある．始めに最長詰め手数が，ランダム配置により設定された局面によって限界が定められる．さらに，簡素化によって駒の数が減る，または弱体化していくといずれこれ以上簡素化できないという局面にたどり着く．すると，アルゴリズムの限界として，それいじょう長い詰め手数の問題を作成できない．駒の移動，または追加，強化がない以上，このアルゴリズムでは 100 手を超えるような長手数の詰め将棋を作成することは難しい．

広瀬らによる手法では逆算によって詰め将棋を作成した．この手法は，逆算を繰り返していくことで非常に長手数の詰め将棋が作成できるように思えるが，現実 9 手より長手数の詰め将棋の作成は現実的でないとされている．その理由として，将棋がチェスなどとは違い相手の駒を取って使用できるというルールが挙げられる．歩のみが存在する局面において，一手前の局面とは歩が後ろに一つ下がった局面である．従って，一手前の局面は 1 つしか存在しないように思えるが，実際には 14 の局面が存在する．これは，今の局面が歩で相手駒を取ったという局面だったと想定すると，歩を手前に下げたとき，歩のいた位置に相手駒を置く必要があるためである．これにより，駒の種類分だけ一手前の局面が生成されるため，逆算を繰り返し駒数が増えてくると一手前の局面数が爆発してしまうのである．従って，9 手程度ならば自由な問題作成が行える一方で，長手数詰め問題を作成することは難しい．

以上のように，現在でも 100 手を超えるような長手数の詰め将棋問題自動創作には至っていない．

4.4.2　面白さについて

自動創作によって作成される詰め将棋は，人が作成したものに比べて面白みが少ないという意見がある．野下のシステムでは作成した詰め将棋をプログラムが解く時間について注目し，多く時間がかかったものを難易度の高い問題として他の問題と区別している．それでも，このような意見が出る理由としては次のような原因があると考える．野下のアルゴリズムで作成される詰め将棋は，簡素なものになりやすいという性質がある．すなわち，駒を弱くするか減らしていくという操作により，盤上から強い駒がどんどん失われるのである．これにより，ランダム配置で複雑な初期局面を得られても，その複雑さが残りにくい．また，ランダム配置において詰み局面を得るために，玉の位置や配置する駒の範囲の限定を行っている．これにより，できあがった局面はいかにも玉方が詰みそうな状況から始まるため，難易度の限界が打ち止めになっているように考える．

逆算によって得られる詰め将棋は，9 手という手数も問題ながら，非常に解答手順が一本道の問題が多い．逆算から得られる詰め将棋は，本来非常に自由であるはずだが，この

ような問題が発生するのは，与える初期局面に原因がある．本来詰め将棋問題は，最初から最後まで動かず，単に他の解答手順を潰すためだけに存在する駒など，多くの駒が配置されているものである．しかし，詰んだ局面を用意した時点では，それらの駒が必要かどうかはまったく分からない．従って，逆算をしていく中でこういった駒を配置しなければならないのだが，ただでさえ逆算の局面数が膨大な中で，手順とは関係のない駒をおいて検証を行うという行為は不可能に近い．このような問題から結局，解答手順を惑わせるような手などが作成できず，非常に簡素な問題になってしまうと考える．

以上のように，面白さについても詰め手数と同様に今後も課題が残っていると考える．

第5章　証明数を用いた詰め将棋問題の特徴

5.1　詰め将棋の面白さ評価について

詰め将棋の面白さについては様々な見方がある．詰め手順の難解さを楽しむ人もいれば，局面が特定の形状をしていることを楽しむ人もいる．詰め将棋は詰め手順や形状などで分類分けされており，曲詰という種類では，初形や詰め上がりの図が文字になっていることを楽しみ，煙詰めという種類では，将棋で使用する全ての駒が盤上から消えていく様子を楽しむ．ある人に面白い詰め将棋が別の人にも面白いとは限らない．詰め将棋の楽しみ方は人それぞれであり，非常にあいまいなものである．

その一方で，詰め将棋には万人に共通して感じられる「面白さ」があるのではないかとも思える．全日本詰将棋連盟が制定した看寿賞という賞は詰め将棋に与えられるものの中で最も価値があるものとされている．看寿賞はその年の最も優れた詰め将棋を短編（17手以下），中編（19手〜49手），長編（51手以上）という3つの部門で表彰する．また，詰め将棋パラダイスで行われた詰め将棋コンテスト[17]では5手詰めの詰め将棋問題を集め，読者の投票によって優秀だと思われる作品上位3位の発表が行われた．このように，多くの人が優秀だと認める作品が存在することは，詰め将棋には共通して面白いと思える要素が存在するのではないかと考えることができる．

詰め将棋の面白さとは何によって決まるのかを解析できれば，詰め将棋を創作するにあたり，良い問題を作成する手がかりとなる．本研究では，この詰め将棋の面白さを定量的に解析する手法について提案と実験を行った．

5.1.1　小山らによる手法と問題点

詰将棋問題の定量的解析については既に小山ら[14]によって研究が行われている．詰め将棋における要素を洗い出し，優秀と呼ばれる作品と一般作品群との傾向の違いについて研究を行っている．例えば，駒数や玉の移動回数などの要素について統計を取り比較を行っている．この研究の中で本研究と最も関連があるのは紛れと変化の数についての部分である.「紛れ」とは，攻め手が選べる王手のかけ方の数であり,「変化」とは受け手が選べる逃げ方の数である．これらの総数が多いほど難解な問題になるとして，詰将棋解答プ

ログラムを利用した際の探索ノード数や探索時間について調査を行い，その値について比較を行っている．

しかし，探索ノード数や探索時間では，攻め手と受け手を一緒にしてしまう点や，探索中の値変化が不明となってしまう問題がある．また，複数の要素で比較を行っているため，ある詰め将棋を単純に評価できない．

5.1.2 提案手法

本研究では，探索アルゴリズムの一つである PNS および DFPN を利用し，探索中の証明数・反証数に着目することで詰将棋問題の評価について調査を行う．証明数・反証数については後述するが，攻め手と受け手が考えなければならない局面数を分離することができ，より詳細な調査を行えると考える．また，証明数・反証数の推移，平均値，最大値などの詰将棋を解いている最中のデータにも着目し，詰将棋問題の評価との関連を調べる．

証明数・反証数から得られる詰め将棋の特徴

PNS において説明したとおり，証明数，反証数とはルートノードの値を確定するのに調べなければならないノードの数であった．ここでは詰め将棋を解く際における証明数，反証数について別の見方を提案する．

証明数はノードを True にするために値を確定させなければばらならない末端ノードの数を表す．証明数が大きい場合，調べなければならないノード数が多いことを表し，反対に証明数が小さい場合は調べるノード数が少ないことを示す．詰め将棋を解く際の証明数は，実際に人間が解く際にも同じように調べなければならない紛れや変化の先の局面数と同じである．これは詰め将棋の解手順が 1 つしか存在しないため，それ以外の手筋が不詰みであることを証明しなければならないことに基づく．従って証明数は，詰め将棋問題の難易度と大きな関係があると考えるが，実際に証明数が 100 という大きな値になったとき，人間は全ての局面を調べるわけではなくある程度あたりや経験を用いて有望な手を探索する．従って，証明数がそのまま難易度になるわけではないが，調べる必要のある局面が多いという状況は，少ないものに比べて難しい問題であるという印象を与えるはずである．

反証数はノードを False にするために値を確定させなければならない末端ノードの数を表す．証明数と同じく，反証数が大きい場合は調べるノード数も多く，反対に小さければ調べるノード数も少なくなる．詰め将棋において反証数は最終的に ∞ となることが決まっている．これは，詰め将棋に必ず詰み手順が存在するため，PNS ないし DFPN で解いた場合に，玉方の詰みが見つかりルートノードの値が False となるためである．従って，反証数は探索の開始から徐々に大きくなり，解手順が見つかった時点で ∞ となる．では，この ∞ 直前の反証数は何を表すのかを考える．この ∞ 直前での反証数は，単に玉方が詰みから逃げるために調べ無ければならないノード数を表している．すると反証数が少ない場合，例えば 1 であった場合，玉方は常に相手の詰みから逃れられる可能性を持っていること

となる．反証数が大きいと玉方が調べなければならないノード数が多く，詰みから逃れるのは容易でないことが窺える．従って，反証数は玉方の逃げられる可能性を表し，これは攻め方から見れば玉方の形勢を表していると考える．詰め手順が分かる直前まで反証数の少ない詰め将棋は，一見すると玉方が詰みそうにないような印象を与えるのではないだろうか．ただし，これは証明数の難易度との関わりより曖昧なものであるため，ここでは単なる仮説としておく．

5.2 実験手法

5.2.1 証明数・反証数の取得手法

詰め将棋を解いた際の証明数は 0，反証数は ∞ となるため，単に解いた後の値を得ても意味は無い．そこで，探索を行っている最中の証明数，反証数に注目しデータを集める．この証明数，反証数の値の取得方法は探索手法に依存しており，PNS なら更新したルートノードの値，反証数なら値を取得するための新しい仕組みを導入する必要がある．ここでは，実際に値を取得するのに使用した手法について解説する．

Proof-Number Search を用いた場合の手法

PNS を利用し探索中の証明数，反証数を取得する方法は非常にシンプルである．PNS はその探索の特性上，末端ノードまで探索，展開した後ルートノードまで証明数，反証数を更新する．従って，ルートノードの値を更新したタイミングで証明数，反証数を記録していけば，初期局面における証明数，反証数の遷移を解析できる．従って，PNS をそのまま使用することで，探索中の証明数，反証数を得ることができる．

Depth-First Proof-Number Search を用いた場合の手法

DFPN を利用して探索中の証明数，反証数を取得するには，専用に仕組みを導入する必要がある．探索手法解説で述べたように，DFPN では一定の深さに留まりながら探索を行い，値の更新は現在見ているノードでしか行われない．ゲーム木全体のノードはルートノードで得るのが良いが，DFPN ではルートノードに戻ってくる回数が PNS に比べて非常に少ないため，そのままでは値が飛び飛びになり利用しづらい．そこで各ノードに計算用の証明数と反証数を持たせ，末端ノードを展開した際にルートノードまでそれら値を更新する仕組みを導入した．具体的には，ノード N を展開した際に N の計算用証明数，反証数の値を計算し設定する．その後，N の親ノードの計算用証明数，反証数を算出し，さらにその親ノードのという具合にルートノードまで値を更新する．計算用証明数，反証数はこのルートノードの値を得るためだけに使用され，探索には影響しない．この仕組みにより，DFPN を利用しながら PNS とほとんど同じデータを取得できる．

手法の違いについて

実験の結果，PNS と DFPN で得られた値にはほとんど差はみられなかった．証明数，反証数が非常に大きな値となる場合に少々差が出るが，ほとんどの場合は完全に一致した．この値の差は PNS が最良優先探索に比べ，DFPN は子ノードを展開してそのノードの有望さを考えながら反復的な深さ優先探索するため，PNS より展開するノードが少し多くなることが原因である．どちらの手法で値を得るべきなのかここでは決めることが難しいため，本論文では PNS によって得られた値に焦点を当てて考察する．

5.2.2 使用する詰め将棋問題について

本実験で使用する詰め将棋問題は，詰め将棋世界誌にて実施された 5 手詰め，7 手詰めコンテスト [17],[16]，看寿賞受賞作品 [21]，詰め将棋問題集に掲載の問題 [18],[19]，また野下および広瀬らの自動生成問題を使用する．

5 手詰め，7 手詰め問題コンテストについて簡単に説明しておく．このコンテストは始めに将棋世界という雑誌において，一般読者より規程詰め手数の詰将棋問題を集った．応募された大量の問題の中からプロ棋士によって大賞候補が絞られ，7 手詰めは 173 題の中から 30 題，5 手詰めは 400 題の中から 40 題が選ばれた．選ばれた作品は再び作者が誰であるか分からないようにした上で，将棋世界誌に掲載され，各問題の解答と読者の考える評価を 1 位から 3 位まで書いたはがきの応募を募った．最終的に読者によって投稿された評価によって問題のランキングが決定し，上位入賞問題が発表された．

看寿賞は詰将棋パラダイス誌にて発表される年間で最も優秀だとされる作品に送られる賞である．短編，中編，長編の三部門に分かれて受賞があり，年度によっては受賞作品なしの部門もある．詰将棋に与えられる賞の中では特に価値があるものとされている．

コンテスト作品や看寿賞作品は優秀な詰め将棋と認められた詰め将棋であり，詰め将棋問題集に載っているような一般的な問題とは一線を画す．従って，これら問題集の問題と優秀な作品を比較することで，面白さには何が関係しているのかを調べる．また，自動生成した問題についても調査を行い，それぞれの創作手法で作成された問題と面白さの関係について調べる．

5.3 実験結果

5.3.1 結果の見方

問題番号は掲載誌において割り振られている番号と同じものを載せてある．PN 最大は，証明数最大を表し，探索中で記録した最大の証明数を記載している．同じく DN 最大は，反証数歳代を表し，詰み手順発見時を除いた探索中の最大の反証数を記載した．更新数はルートノードの値を更新した回数で，末端ノードを展開した回数と等しい．PN 平均は証

明数平均を表し，ルートノードに更新するたびに証明数を加算していき，最終的にそれを更新数で割ったものである．一回の探索において平均的にどれほどの証明数となっているのかを表している．DN 平均は PN 平均と同じく反証数平均を表し，こちらも合計値を更新数で割ったものを表示している．

証明数が問題の難易度を表すのならば，証明数最大はその問題において最も難しいと感じた瞬間を表すと考える．反証数最大は，普通詰み手順発見直前の値であり詰み手順が発見されそうなとき，玉方にはどれほど逃げ切れるチャンスがあったかを表していると考える．反証数最大が小さいほど，最後まで気の抜けない詰め将棋と感じるのではないだろうか．ただし，この論文ではそれを証明することはできないため，単なる一情報として示した．証明数平均は，平均してどれ程の証明数を持って各探索を行っているのかを表している．従って，証明数平均は問題を解いている間に感じる平均的な難易度であると考える．平均値が高ければ，問題を解くために非常に試行錯誤しなければならず，反対に値が低ければ紛れや変化が表れても，簡単にその手筋が解答でないことを証明できるのではないだろうか．反証数平均は，各探索において平均的に玉方が逃げ切れるチャンスを表していると考える．値が低ければ，常に玉方にはチャンスがあるように見えるため気が抜けない問題になっているのではないだろうか．こちらも，反証数最大と同じくここでは証明できないため参考データとして掲載している．

5.3.2 コンテスト問題について

始めにコンテスト問題の値について考察する．表 5.3.2 に 7 手詰めコンテスト問題，表 5.3.2 に 5 手詰めコンテスト問題の実験結果を示す．

表 5.1: 7 手詰めコンテスト上位問題

問題順位	PN 最大	DN 最大	更新数	PN 平均	DN 平均
1 位	116	3258	235574	71.630	2252.287
2 位	32	3985	58276	21.515	2576.343
3 位	3	111	206	2.194	65.049

表 5.2: 5 手詰めコンテスト上位問題

順位	PN 最大	DN 最大	更新数	PN 平均	DN 平均
1 位	16	3812	30369	10.580	2504.111
2 位	593	6587	2989494	335.244	359.172
3 位	34	2827	55154	23.245	1829.251

はじめにPN,DN最大に着目しデータを見てみる．まずPN最大についてだが，表5.3.2をみると，大賞から順位に従って減少していることが分かる．証明数が難易度を表すのであれば，PN最大は問題を解いている最中の最も難しく感じた瞬間（最高難易度）を表していると考える．従って，単純に最高難易度によってコンテストの結果が決まったように見えるが，3位作品のPN最大は3であり，これは一般詰将棋問題の平均値より下である．問題における最高難易度だけがコンテスト結果につながった訳ではないことは，表5.3.2においても示しており，大賞作品のPN最大値は他2作品より低い値となっている．次にDN最大だが，反証数は証明数や更新数によって値の変動が縛られるため，この値だけを見ても他問題との形勢を比較することは難しい．

次にPN,DN平均に着目しデータを見てみる．まずPN平均についてだが，PN最大と同じく表5.3.2において大賞から順位に従って減少している．PN平均は，問題を解いている最中に平均的に感じた難易度を表すと考える．しかし，これもPN最大と同じように3位作品は一般平均よりPN平均が低く，表5.3.2においても順位との関連性が見られない．次にDN平均を見てみる．DN平均は，問題を解いている最中に感じる平均的な形勢を表し，値が低いほど形勢が拮抗しているように見えると考える．表5.3.2においては大賞作品の方が，2位作品よりDN平均が少ない．また，3位作品のDN平均は一般作品の平均値よりぐっと下になっている．3位作品は難易度でみれば，それほど難しい問題ではないが，見た目の形勢は非常に緊迫したものに感じ取れるのではないだろうか．その一方で表5.3.2の方では表5.3.2とは異なり，DN平均と順位の関連性は見つけづらい．

表5.3.2において各値と順位との関連性が感じられづらいのは，無駄合いによる証明数・反証数の増加が原因だと考える．無駄合いは，攻め手が飛車や角などの飛駒で玉を遠距離から王手した場合に，受け手が持駒から合駒を使って玉を守った結果，その合駒が攻め手の持駒に残ったまま受け手が詰んだとき，合駒した手のことを無駄合いと呼ぶ．本実験では，人間が詰将棋問題を解く場合，ある手が無駄合いかどうかはすぐに分からないという仮定の元実験しているため，無駄合いが起きやすかった問題，特に5手詰めコンテスト2位作品の値だけが突出している．無駄合いを考える際の我々の思考方法についてはより正確に議論する必要があり，さらなる正しいデータ取得方法を見つけられれば，各値に順位とのより深い関連性が見つけられるのではないかと期待する．

5.3.3　一般問題誌の問題について

表5.3.3に一般問題誌の7手詰め問題を，表5.3.3に一般問題誌の5手詰め問題を示す．

始めにPN，DN最大について見てみる．5手詰め，7手詰め問題のPN最大は，コンテスト問題に比べ小さい値となっている．一部はコンテスト問題より大きな値となっているものもあるが，コンテスト問題において特に値が大きかったものと比べると非常に小さい値である．使用した一般問題誌では実践力を鍛える目的で問題を作成したと書かれており，パズルとしての複雑さを求めた問題ではないことが窺える．5手詰めと7手詰めでは5手詰め問題の方が全体的に証明数が高めだが，問題作成の意図や問題誌の傾向にも依存

するため，一概に何かを言うことはできない．なお，7 手詰め問題の方が PN 最大，DN 最大は大きくなる．これは 7 手詰めに 5 手詰めが内包されていることを想像するとわかりやすい．DN 最大について見てみると，表 5.3.3 の問題番号 21,25 において証明数は同じだが，反証数が非常に離れていることが分かる．同じ 7 手詰めでもこれほど異なるのは面白い．同じように 5 手詰めでも問題番号 2 と 4 で反証数が大きく離れている．これらの問題について解析を行えば DN 最大が見た目の形勢を表すかどうかについて判断できると思われるが，ここでは単に値についてのみ言及しておく．

次に，PN，DN 平均について見てみる．これら平均値はコンテスト問題に比べとても低い値になっている．特に，5 手詰めに至っては，PN 平均が 3 位作品より高い値となったものが存在しない．詰め将棋の評価において必要なのは瞬間的な難易度よりも平均して感じる難易度なのだろうか．表 5.3.3 において問題番号 22 の PN 平均が 1 となっている．これは，この問題に考えるべき紛れや変化が非常に少なく，解答手順が一本道に近い問題であることを示している．また，偶然かもしれないが同じ PN 最大の問題において PN 平均の値がは DN 最大の小さい問題の方が若干大きい．こちらも難易度や面白さと関係あるのか調査が必要である．

5.3.4 自動生成問題について

表 5.3.4 に野下の手法によって自動生成された 15 手詰め問題，表 5.3.4 に広瀬らの手法による自動生成問題，表 5.3.4 に看寿賞受賞作品の内 15 手詰めだった問題の実験結果を示す．

始めに表 5.3.4 の野下による自動生成問題を見てみる．野下の自動生成問題を比較するために，看寿賞を受賞した問題の中から 15 手詰めだったものを抜き出し，実験を行った．両者を比較すると，PN 最大についてみれば，いくつかは看寿賞受賞作品とほとんど変わらない値であり，一部の問題は看寿賞作品より PN 最大が大きい．DN 最大については全体的に看寿賞作品の方が大きくなっている．PN 平均について見ると，PN 最大が大きいもの以外は看寿賞作品より小さな値となっている．DN 平均は全てにおいて看寿賞作品の方が大きい．このように見ると，野下の自動生成問題は看寿賞作品と同程度，または優秀であるかのように言えそうだが，更新数に着目すると野下の問題番号 5 以外は看寿賞作品

表 5.3: 一般問題誌 [19] の 7 手詰め問題

問題番号	PN 最大	DN 最大	更新数	PN 平均	DN 平均
21	9	916	5579	6.513	556.034
22	1	72	90	1.000	42.144
23	3	198	429	2.613	110.163
24	2	111	226	1.841	66.142
25	9	2644	12756	6.497	1477.837

表 5.4: 一般問題誌 [18] の 5 手詰め問題

問題番号	PN 最大	DN 最大	更新数	PN 平均	DN 平均
1	8	457	2071	5.705	275.024
2	12	213	1578	8.590	142.575
3	5	224	1023	3.929	128.341
4	12	2072	10622	7.721	1351.236
5	9	375	2344	5.985	214.409

表 5.5: 野下による 15 手詰め自動生成問題

問題番号	PN 最大	DN 最大	更新数	PN 平均	DN 平均
1	23	154	2476	13.491	74.509
2	39	3624	61715	24.523	2291.601
3	127	15684	718674	75.241	1494.060
4	10	361	2248	6.363	172.194
5	130	46003	3989896	80.282	269.115

表 5.6: 広瀬らによる自動生成問題

問題番号	詰手数	PN 最大	DN 最大	更新数	PN 平均	DN 平均
1	3	2	65	87	1.046	41.793
2	5	4	243	550	2.655	136.802
3	5	2	46	72	1.375	27.319
4	5	3	111	274	2.183	58.518
5	7	8	329	1637	5.260	195.294
6	9	288	135	12740	120.875	42.899

表 5.7: 看寿賞受賞の 15 手詰め問題

受賞年度	PN 最大	DN 最大	更新数	PN 平均	DN 平均
10 年度	46	41317	1019556	32.687	26993.076
13 年度	49	36351	1015886	31.502	20553.620
17 年度	66	115586	4360117	46.344	61227.917
23 年度	25	61781	739489	16.919	38763.192

より非常に少ない．看寿賞の更新数が 100 万を超えるような問題において，PN 平均が 30 を超えているような場合，これだけの探索の間常に PN 平均値が高いというのは問題の平均難易度に加え，問題を解く時間が加わっているように考える．野下の問題番号 1 と看寿賞受賞の 23 年度の問題は PN 最大，PN 平均はほとんど同じだが，更新数に大きな開きがある．つまり，これら二つの問題は感じる平均難易度も瞬間的な最大難易度が同じだが，答えを導くのに必要な時間が看寿賞作品の方が圧倒的に長いと考える．したがって，野下の問題番号 5 は看寿賞に選ばれる可能性がある一方で，それ以外の問題はそれほど優秀でないのかもしれない．野下の自動生成問題は論文の付録より抜粋したものだが，野下によれば数ヶ月システムを稼働させ生成された大量の問題の内，解くのに時間がかかった問題を抜き出すことによって良い問題を選別しているそうである．論文に掲載されている問題がどのような基準で書かれているのかは分からないが，野下によってさらに選別されたものであった場合，自動生成問題ではあるが人間の評価を一度通したものとなる．従って，この表から野下のアルゴリズムが面白い問題を優先的に生成しているかどうかを述べることはできない．ランダム配置に依存しているため，面白い問題を狙って生成することはできないが，長時間稼働させればこのような問題も生成できるという点ではこの自動生成手法は良い手段であると考える．また，野下の自動生成問題のいくつかは詰め将棋誌にて紹介された経歴もあり，この手法の大きな可能性を感じさせる．

次に広瀬らの自動生成問題を表 5.3.4 に示し考察を行う．論文に掲載された付録の問題を実験したが，問題数が少なく詰め手数がばらばらであったため，詰め手数が統一されていない．こちらは全体を通して非常に値が小さい．詰め手数が短いことも原因だが，同じ 5 手詰めでも一般問題誌より値が小さいのはこの手法に問題がある可能性を示している．広瀬の問題番号 6 は PN 最大などが非常に大きいが，このように DN 最大より PN 最大が非常に大きくなるのは無駄合いが発生しているときの特徴である．論文付属の問題が少ないためこの手法について評価することは難しいが，広瀬らの手法で証明数が大きな問題を作成することは難しいのかもしれない．

5.3.5　証明数の推移グラフについて

本システムではルートノードの値が更新されるたびに証明数，反証数を記録しているが，この証明数を縦軸に，更新数を横軸に取り，探索中の証明数の推移を表すグラフを作成した．図 5.1, 図 5.2 にコンテストの上位 3 位問題における証明数の推移を示す．

縦軸が証明数，横軸が更新回数である．ここでは，形状を見やすいようにスケール幅をそろえてあるため，最大値等の値が異なる点に注意して頂きたい．

証明数の推移は，探索中において証明しなければならないノード数が増加していく様子が見て取れる．このグラフは，詰将棋問題を解く際に人間の頭の中で感じられる難易度がどのように変化しているのかを表すと考える．図を見ると，コンテスト上位問題は一定の難易度まで証明数が上昇した後，突然値が下がる．詰将棋問題を解いている場合，一定時間悩んだ後，急にひらめき問題が解けることがあるが，このグラフはまさにそういっ

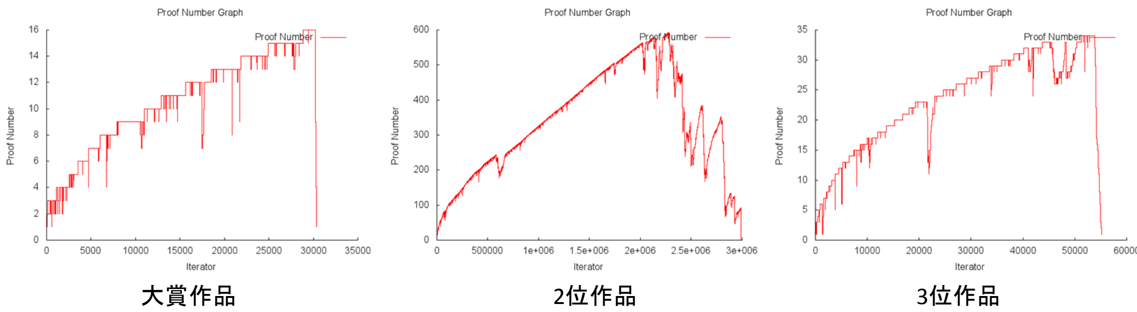

図 5.1: 5 手詰めコンテスト上位 3 位問題の証明数推移

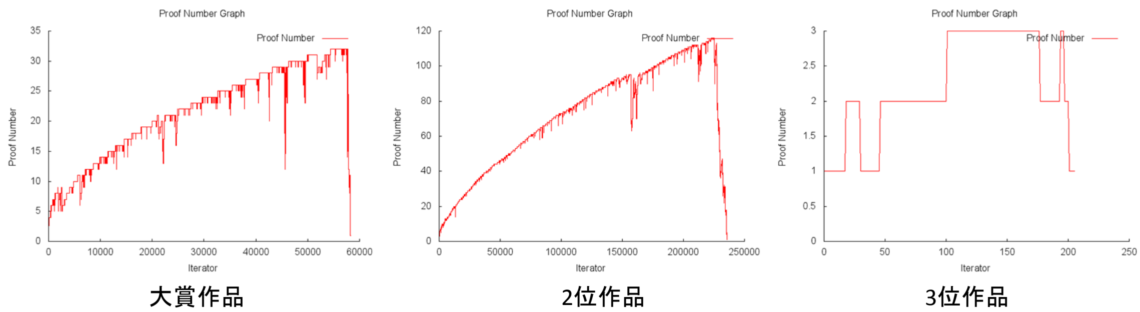

図 5.2: 7 手詰めコンテスト上位 3 位問題の証明数推移

たことを表しているのではないだろうか．コンテスト上位問題は基本的に，同じような形状を取っているが，多くの一般的な詰将棋問題は図 5.2 の 3 位問題のような形状を取っている．この図に多く現れる山なりに証明数が上昇した後，急激に値が下がるというのは，コンテスト上位問題に共通して現れる特徴的な形であることを述べておく．

第6章　まとめ

6.1　実験結果について

本論文および実験において示したことは以下の通りである．

- 証明数・反証数の新しい見方として，詰め将棋問題の評価に使用するという考えを示した．
- 評価に利用する証明数・反証数の取得方法について提案した．
- 取得した証明数・反証数は詰将棋問題の評価に一定の関わりがある可能性を示した．

得られた証明数・反証数のデータから，難易度や見た目の形勢が詰将棋問題の評価と関わっていることは示したが，コンテストの順位はこれらの数値だけでは成り立ってはいなかった．これは小山らの研究において，多くの要素が詰将棋問題の芸術性に影響を与えているという結果の裏付けとなったと考える．ただし，証明数・反証数は，その値だけで詰将棋問題における多くの情報を持っていることから，解答手数に縛られている証明数・反証数の正規化や，無駄合いの正確な処理を行うことで，将来的には単一の数値だけで多くの詰将棋問題の評価を行えるようになると考える．その先駆けとして，本論文ではデータの取得方法提案およびデータの考察を行った．

6.2　実験結果と自動創作手法について

自動創作手法については大きく分けて野下によるランダム配置を利用した手法と，広瀬らによる逆算による手法があった．実験により次のようなことが分かった．

- 野下による手法は問題を大量作成することによって優秀な作品を創作できる可能性がある
- 広瀬らによる手法では詰め手数を伸ばすことが難しく，証明数も大きくなりずらい
- どちらの手法も面白い問題を狙って作成することが難しい

野下の手法では実験によって，作成された問題が優秀である可能性を示した．しかし，これらの問題は大量の自動生成によって生まれた中から抜粋されたものであった．また，20 手を超える詰め将棋を作成できる一方で，最大詰め手数には限界があることも述べた．詰め手数が長く，面白いと思える詰将棋を作成するためには，野下の手法をそのまま使用することはできないだろう．

広瀬らの手法で証明数を大きくしずらいのは，逆算の不可を下げるためだと考える．無駄合いが発生するような飛び駒での王手を行っていた局面の一手前を作成することは非常に困難であり，結果的に飛び駒での王手などは敬遠せざるをえないのではないだろうか．逆算を利用すれば，作成したいと思った特徴を持った詰将棋を作成できるが，このように逆算という処理に負荷が多くかかるため，詰め手数を伸ばすことや面白い問題を自由に作成することは難しいだろう．

従って，現在存在する自動生成手法を参考にした上で，あらたな自動生成手法を考案する必要があると考える．

6.3 今後の方針

今後研究を継続していく上で必要な懸案事項を以下に示す．

- 無駄合いの取り扱い検討
- 証明数，反証数が仮説通り難易度や形勢を表すかの確認
- 解答時間についての検討
- 面白い問題を狙って生成できる手法の考案

無駄合いについてはより考慮が必要であり，この値にどのような補正をかけるかは今後さらなる検討が必要である．簡単な無駄合いの場合，人間には即座に理解できる場合があり，そのような場合 PN 最大等の値の意味が失われる．特に，1 手積み問題で飛び駒を利用し王手をかけるものでは，PN 最大が非常に大きな値になるが，人間に取ってみればほとんど考える必要がなく，考える無ければならない局面はごくわずかである．ただし初心者にしてみれば，無駄合いかどうかすぐに判別することは難しく，本システムと同じように合駒を多く考慮してしまう可能性もある．無駄合いについて詰め将棋を解く人間がどのような考え方をしているかより詳細に調べる必要があるだろう．

本研究では証明数，反証数が詰め将棋の評価に一定の関わりがあること示したが，本当に証明数が難易度を表し，反証数が形勢を表すのかについては示せていない．これら二つの値が正確に何を表すのか確認することは大きな課題の一つである．

解答時間についての検討は，自動生成問題の実験考察で述べたように，PN 最大や PN 平均だけでは問題の難易度を完全には表せないのではないかという考察から得られた課題である．平均的な難易度や，一瞬感じる最も難しい瞬間が同じ問題でも，解くために必

要な時間が異なれば難易度は違うのではないだろうか．これをどのように数値化し比較するのか検討する必要がある．

最後におもしろ問題を作成する自動創作アルゴリズムの開発である．本実験は全てこのためのものであり，自動創作アルゴリズムにおいて面白い問題を作成するにはどのような方針でアルゴリズムを組めば良いかを見定めるためのものであった．実際，証明数，反証数が評価に一定の関わりを持っていることを発見でき，この証明数，反証数を基準にした創作アルゴリズムを考案していくことが今後の課題となる．作成した問題が面白いと判断されれば，この証明数，反証数が評価と関係しているということの裏付けにもなると考える．

以上のような事項を博士課程で引き続き研究していきたい．

第7章　謝辞

本研究を進めるにあたり，ご指導を頂いた卒業論文指導教員の飯田弘之教授に感謝致します．また，日常の議論を通じて多くの知識や示唆を頂いた飯田・池田研究室の皆様に感謝します.

参考文献

[1] 伊藤琢巳, 最良優先探索によって詰め将棋を解くプログラム, コンピュータの進歩, pp71-89, 共立出版, 1996.

[2] Akihiro Kishimoto, Mark H.M. Winands, Martin Müller, Jahn-Takeshi Saito. GAME-TREE SEARCH USING PROOF NUMBERS: THE FIRST TWENTY YEARS, ICGA Journal 2012 no.3.

[3] 野下浩平, 詰め将棋を解くプログラム T2, コンピュータの進歩, pp.50-70, 共立出版, 1996.

[4] T. Hashimoto. Searching for Solutions in Complex Games: Shogi Endgame and Amazons. PhD thesis, Shizuoka University.

[5] A. Kishimoto. Correct and Efficient Search Algorithm in the Presence of Repetitions. PhD thesis, University of Alberta, 2005.

[6] Knuth, D. E. and Moore, R. W. (1975). An analysis of alpha-beta pruning. Artificial Intelligence, Vol. 6, No. 4, pp. 293-326.

[7] A. Nagai. Df-pn Algorithm for Searching AND/OR Trees and Its Applications. PhD thesis, Department of Information Science, University of Tokyo, 2002.

[8] D. A. McAllester. Conspiracy number for min-max search. Artificial Intelligence, 35;287-310, 1988.

[9] M. Seo, H. Iida, and J. W. H. M. Uiterwijk. The PN*-search algorithm: Application to tsume-shogi. Artificail Intelligence, 129(1-2):253-277, 2001.

[10] L. V. Allis, M. van der Meulen, and H. J. van den Herik. Proof-number search. *Artificial Intelligence*, 66(1):91-124, 1994.

[11] K. Hashimoto. Micro cosmos. *Tsume-shogi Paradise*, 1986. (in Japanese).

[12] K. Noshita, A Note on Algorithmic Generation of Tsume-Shogi Problems. In *Game Programming Workshop in Japan '96*, pages 27-33, Kanagawa, Japan, 1996.

[13] M. Hirose, T. Ito, and H. Matsubara. Automatic Composition of Tsume-Shogi by Reverse Method. *Journal of Japanese Society for Artificial Intelligence*, 13(3):452-460, 1998.

[14] 小山 謙二, 河野 泰人. 名作詰将棋における感性の定量的評価, 情報処理学会論文誌, 35(11):2338-2346, 1994

[15] C. E. Shannon. Programming a computer for playing chess. Philosophical Magazines, 41:256-275, 1950.

[16] "7 手詰めコンテスト問題", 将棋世界, 1992 年, 3 月号付録

[17] "5 手詰めコンテスト問題", 将棋世界, 1992 年, 9 月号付録

[18] "明解 5 手詰め 39 題", 将棋世界, 1999 年, 5 月号付録

[19] "大五郎の痛快 5 手 7 手", 将棋世界, 1997, 8 月号付録

[20] "中田章道短編傑作選", 将棋世界, 2003, 7 月号付録

[21] 「詰将棋パラダイス 看寿賞」, http://www005.upp.so-net.ne.jp/tsumepara/contents/4appre/kanju/kanju0.htm

[22] 石飛 太一, 飯田弘之. 詰将棋問題の感性評価と証明数に関する考察, *Game Programming Workshop in Japan 2012*, pages 163-166, Hakone, Japan, 2012.

付録（詰め将棋データ）

表 7.1: 一般問題誌 [18] の 5 手詰め問題

問題番号	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
1	8	457	2071	11816	569574	5.705	275.024
2	12	213	1578	13555	224984	8.590	142.575
3	5	224	1023	4019	131293	3.929	128.341
4	12	2072	10622	82011	14352830	7.721	1351.236
5	9	375	2344	14028	502575	5.985	214.409
6	4	843	2339	7077	1072640	3.026	458.589
7	9	2556	13025	75242	21899895	5.777	1681.374
8	3	109	244	600	13383	2.459	54.848
9	13	2433	17994	164393	27558058	9.136	1531.514
10	8	2211	8527	42879	11470484	5.029	1345.196
11	4	222	498	1200	60303	2.410	121.090
12	12	1851	11530	90489	12356784	7.848	1071.707
13	5	382	1099	3666	253293	3.336	230.476
14	7	1242	4826	23521	3953140	4.874	819.134
15	4	371	928	2476	204906	2.668	220.804
16	9	976	4908	28828	3039254	5.874	619.245
17	4	332	986	3341	184435	3.388	187.054
18	3	155	303	625	26760	2.063	88.317
19	4	377	1253	4000	293199	3.192	233.998
20	3	652	1608	3831	564958	2.382	351.342
21	4	471	1003	2650	250667	2.642	249.917
22	6	904	2839	12177	1532504	4.289	539.804
23	1	110	128	128	8406	1.000	65.672
24	3	53	84	135	2652	1.607	31.571
25	3	274	502	1094	83625	2.179	166.584
26	7	423	1563	7389	395880	4.727	253.282
27	4	385	1229	3466	267680	2.820	217.803
28	8	704	3318	16650	1432663	5.018	431.785
29	3	165	316	689	30066	2.180	95.146
30	2	137	240	429	19499	1.788	81.246
31	4	151	502	1419	43776	2.827	87.203
32	9	613	3238	18346	1191778	5.666	368.060
33	3	271	515	1044	75331	2.027	146.274
34	3	472	1070	2453	263610	2.293	246.364
35	9	1028	5079	31017	3444040	6.107	678.094
36	4	758	1728	5121	728134	2.964	421.374
37	1	29	15	15	345	1.000	23.000
38	3	179	321	700	30349	2.181	94.545
39	6	940	3365	14079	1847080	4.184	548.909

表 7.2: 5 手詰めコンテストの問題

問題番号		PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
	1	16	772	7883	81054	3398463	10.282	431.113
	2	7	970	3893	19369	2303786	4.975	591.777
	3	2	303	407	684	66981	1.681	164.572
	4	64	66189	4114602	151995532	1073741822	36.941	260.959
	5	193	62477	6766685	823089994	1073741822	121.639	158.681
	6	47	13961	323860	9416855	1073741822	29.077	3315.451
	7	12	693	4620	35058	2203720	7.588	476.996
	8	105	2483	151998	10736013	249336604	70.633	1640.394
	9	7	769	3219	12817	1468699	3.982	456.259
	10	7	176	778	3692	89648	4.746	115.229
(2 位)	11	593	6587	2989494	1002211216	1073741822	335.244	359.172
	12	19	1412	18428	220616	15008015	11.972	814.414
(3 位)	13	34	2827	55154	1282037	100890516	23.245	1829.251
	14	66	150	4523	178072	330859	39.370	73.150
	15	5	341	915	2852	183735	3.117	200.803
	16	5	1870	5544	19854	5997358	3.581	1081.775
	17	11	3603	17434	128890	39877903	7.393	2287.364
	18	2	86	118	158	6122	1.339	51.881
	19	41	2056	57098	1434816	90329835	25.129	1582.014
	20	3	267	457	841	74629	1.840	163.302
	21	36	299	7570	163158	1605157	21.553	212.042
	22	61	3001	50506	1537438	56554944	30.441	1119.767
	23	31	1435	26513	519897	26118140	19.609	985.107
	24	29	696	12430	221199	5676086	17.796	456.644
	25	641	133	73170	23101118	4030527	315.718	55.084
	26	8	888	4042	22894	2079597	5.664	514.497
	27	3	360	696	1568	144667	2.253	207.855
	28	3	25	20	35	240	1.750	12.000
	29	46	346	13509	249790	2598160	18.491	192.328
	30	8	1064	4009	22723	2650235	5.668	661.071
	31	8	540	2234	11888	690887	5.321	309.260
	32	8	540	2234	11888	690887	5.321	309.260
	33	9	204	999	5529	135985	5.535	136.121
	34	5	365	1179	4028	235402	3.416	199.662
	35	2	94	106	129	5928	1.217	55.925
	36	47	1416	39459	1205943	35645665	30.562	903.360
(1 位)	37	16	3812	30369	321308	76047336	10.580	2504.111
	38	8	236	928	5541	140418	5.971	151.313
	39	120	10029	1148486	83649595	1073741822	72.835	934.919
	40	25	3079	38642	627871	85040845	16.248	2200.736

表 7.3: 一般問題誌 [19] の 7 手詰め問題

問題番号	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
21	9	916	5579	36338	3102112	6.513	556.034
22	1	72	90	90	3793	1.000	42.144
23	3	198	429	1121	47260	2.613	110.163
24	2	111	226	416	14948	1.841	66.142
25	9	2644	12756	82874	18851284	6.497	1477.837
26	3	691	1855	4359	709497	2.350	382.478
27	3	53	119	267	3947	2.244	33.168
28	7	270	1649	6944	318689	4.211	193.262
29	9	322	1515	9483	253400	6.259	167.261
30	3	289	621	1320	87580	2.126	141.031
31	3	860	2062	4974	971728	2.412	471.255
32	4	408	1406	3886	306910	2.764	218.286
33	9	2266	11390	67496	16467952	5.926	1445.825
34	8	1713	8362	44529	8889836	5.325	1063.123
35	15	4113	27035	293740	68459825	10.865	2532.267
36	9	300	1229	7364	240965	5.992	196.066
37	53	5852	182416	6295055	773477605	34.509	4240.185
38	2	707	1245	2139	469516	1.718	377.121
39	2	357	592	1056	116740	1.784	197.196

表 7.4: 一般問題誌 [20] の 7 手詰め問題

問題番号	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
2	2	42	67	84	1741	1.254	25.985
3	10	1567	9118	61474	9530923	6.742	1045.287
4	6	397	983	3763	214782	3.828	218.496
5	7	1610	5589	27965	5385477	5.004	963.585
6	8	1090	5522	30098	3501242	5.451	634.053
7	4	570	1627	4992	478890	3.068	294.339
8	2	293	510	774	80623	1.518	158.084
9	36	16570	276713	6762851	1073741822	24.440	3880.345
10	7	389	2084	9942	549779	4.771	263.810
11	9	554	2801	17692	977382	6.316	348.940
12	339	5407	1350538	269367157	1073741822	199.452	795.047
13	8	2706	10102	54037	17133726	5.349	1696.073

表 7.5: 7 手詰めコンテストの問題

問題順位	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
1	116	3258	235574	16874217	530580319	71.630	2252.287
2	32	3985	58276	1253828	150138974	21.515	2576.343
3	3	111	206	452	13400	2.194	65.049

表 7.6: 野下によるランダム配置を用いて生成された問題

問題番号	詰手数	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
1	15	23	154	2476	33403	184485	13.4907	74.5093
2	15	39	3624	61715	1513439	141426165	24.5230	2291.6012
3	15	127	15684	718674	54073737	1073741822	75.2410	1494.0596
4	15	10	361	2248	14304	387093	6.3630	172.1944
5	15	130	46003	3989896	320318738	1073741822	80.2825	269.1152
6	17	20	618	7303	98680	2811775	13.5123	385.0164
7	17	220	3510	251303	23759969	227180709	94.5471	904.0111
8	17	9	1574	11387	74885	10241955	6.5764	899.4428
9	17	36	1409	29793	601096	20692030	20.1757	694.5266
10	17	15	1480	13734	141939	9561145	10.3349	696.1661
11	17	19	6958	61559	655362	183196716	10.6461	2975.9534
12	17	11	305	1606	9377	192925	5.8387	120.1276
13	19	20	3240	37344	471877	58508986	12.6360	1566.7573
14	19	42	3147	129774	3228220	298394935	24.8757	2299.3430
17	17	10	1687	10001	71703	9825139	7.1696	982.4157
18	17	8	251	1610	9666	198201	6.0037	123.1062
19	19	12	588	4944	36602	1807539	7.4033	365.6025
20	25	9	336	2872	16882	604465	5.8781	210.4683

表 7.7: 広瀬らによる逆算法を用いて生成された問題

問題番号	詰手数	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
1	3	2	65	87	91	3636	1.0460	41.7931
2	5	4	243	550	1460	75241	2.6545	136.8018
3	5	2	46	72	99	1967	1.3750	27.3194
4	5	3	111	274	598	16034	2.1825	58.5182
5	7	8	329	1637	8610	319697	5.2596	195.2944
6	9	288	135	12740	1539947	546528	120.8750	42.8986

表 7.8: 看寿賞受賞作品の内 15 手詰めの問題

受賞年度	PN 最大	DN 最大	更新数	PN 合計	DN 合計	PN 平均	DN 平均
10 年度	46	41317	1019556	33326082	27520952874	32.687	26993.076
13 年度	49	36351	1015886	32002887	20880134899	31.502	20553.620
17 年度	66	115586	4360117	202064307	2.66961E+11	46.344	61227.917
23 年度	25	61781	739489	12511599	28664954205	16.919	38763.192